Ejercicio evaluable de normalización

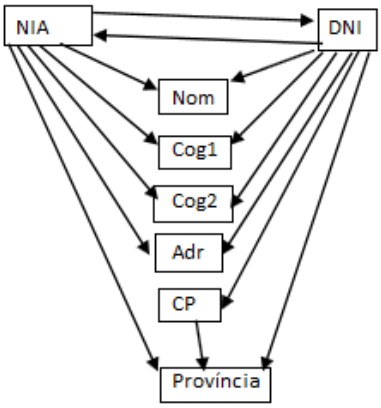
Juan Ignacio López 1ro DAM

ALUMNE(NIA, DNI, Nom, Cog1, Cog2, Adr, CP, Provin)

CP:{ NIA }

UNI:{ DNI }

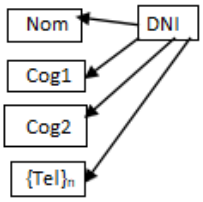
VNN:{ Nom, Cog1, Adr, CP, Provin }



PROFESSOR(DNI, Nom, Cog1, Cog2, {Tel}$_n$)

CP:{ DNI }

VNN:{ Nom, Cog1}

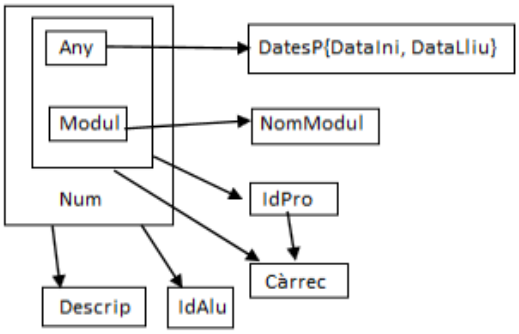


PROJECTE(Any, Modul, Num, DatesP{DataIni, DataLliu}, NomModul, Descrip, IdPro, Càrrec, IdAlu)

CP:{ Any, Modul, Num }

Cal:{ IdPro }->PROFESSOR

Cal:{ IdAlu }->ALUMNE



1FN

La tabla profesor no esta en 1FN ya que no todos sus atributos son atómicos. El atributo teléfono, al ser multivaluado, puede ser dividido en sus diferentes valores a la hora de acceder a él. Para corregir esto creamos otra tabla para contener los teléfonos que se relacione con la tabla profesor, quedando de esta manera:

PROFESSOR(DNI, Nom, Cog1, Cog2)

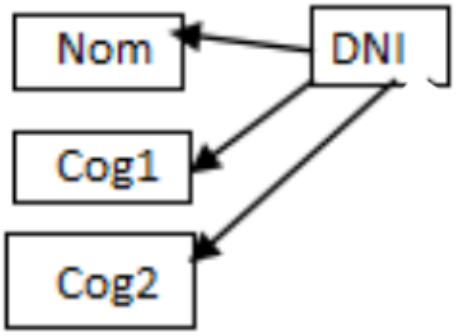
CP:{ DNI }

VNN:{ Nom, Cog1}

TELÉFONOS (IdPRO, Teléfono)

CP: { IdPRO, Teléfono}

CAj: { IdPRO } -> PROFESSOR



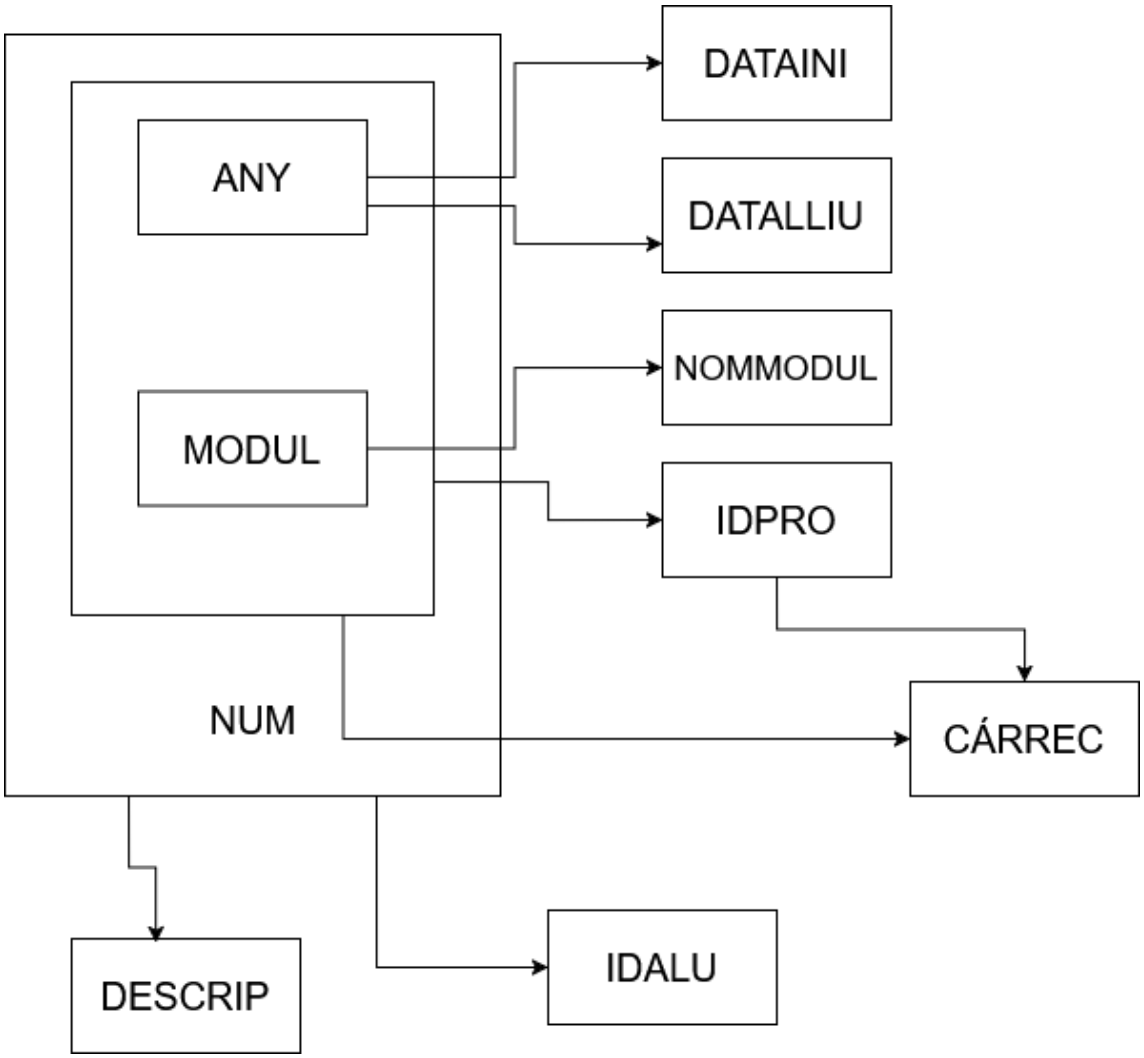
Proyecto tampoco está en 1FN porque DatesP no es atómico debido a que es un dato compuesto y puede ser dividido en DatesIni y Dates Lliu asique creamos una tabla en la que los datos estén desestructurados.

PROJECTE(Any, Modul, Num, DataIni, DataLliu, NomModul, Descrip, IdPro, Càrrec, IdAlu)

CP:{ Any, Modul, Num }

Cal:{ IdPro }->PROFESSOR

Cal:{ IdAlu }->ALUMNE

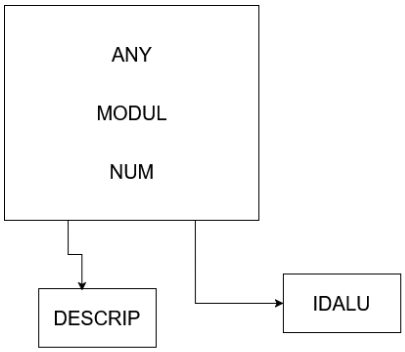


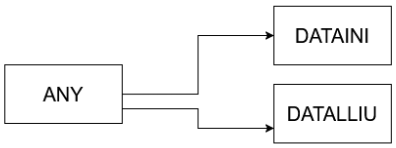
La tabla alumno si esta en 1FN.

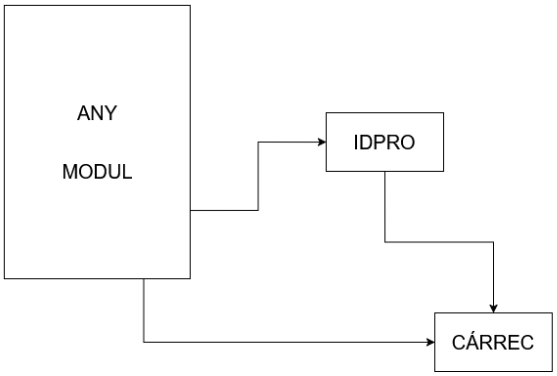
2FN

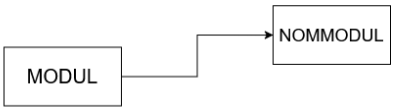
La tabla proyecto no esta en 2FN porque no todos los atributos dependen del conjunto de cp de la tabla, algunos dependen de una sola clave o de 2.

Por ende hay que separarlas en varias tablas, hay que crear una tabla por cada subconjunto de cp a las que dependan cada atributo y mandar a esos atributos a esa tabla. Hay que crear una tabla de módulos, una tabla de años, una tabla de añosxmodulos (a la cual llamaremos “curso”). Nos quedaria así:









PROJECTE(IdAny, IdModul, Num, Descrip,IdAlu)

CP:{ IdAny, Modul, Num }

Cal:{ IdAlu }->ALUMNE

Cal:{ IdAny }->ANY

Cal:{ IdModul }->MODUL

CURSO(IdAny, IdModul, IdPro, Càrrec)

CP:{ IdAny, IdModul }

Cal:{ IdPro }->PROFESSOR

Cal:{ IdAny }->ANY

Cal:{ IdModul }->MODUL

ANY(IdAny, DataIni, DataLliu)

CP:{ IdAny }

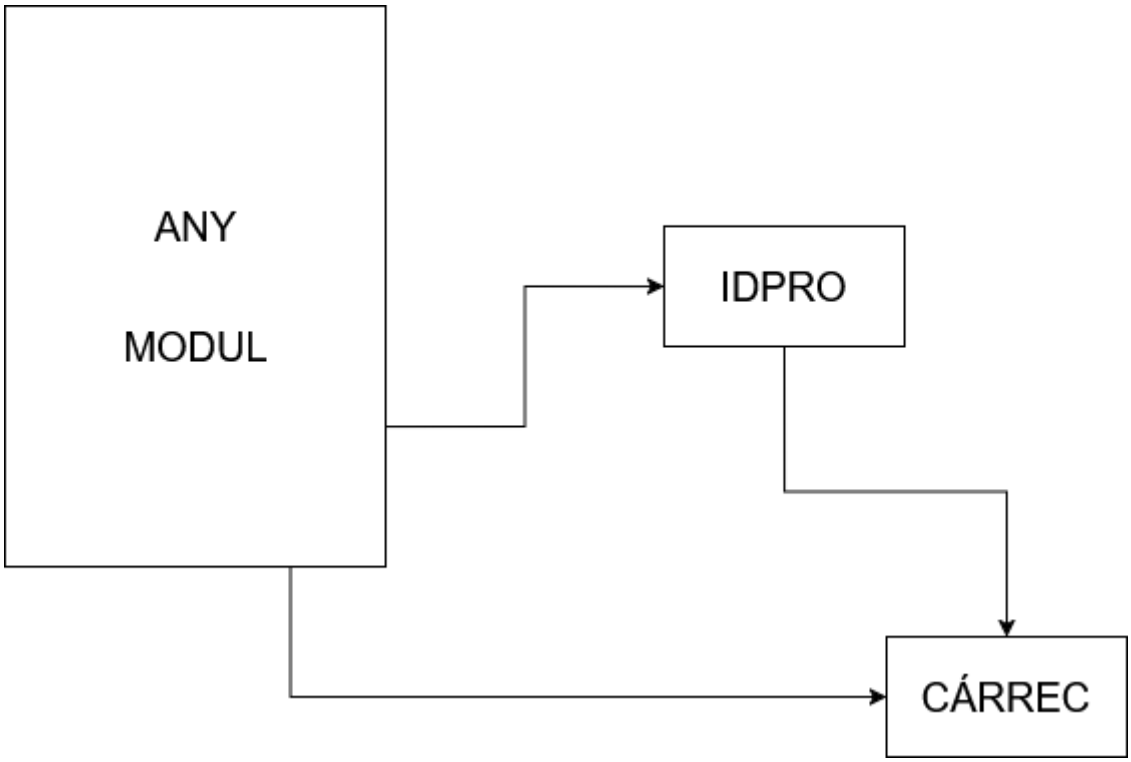
MODUL(IdModul, NomModul)

CP:{ IdModul }

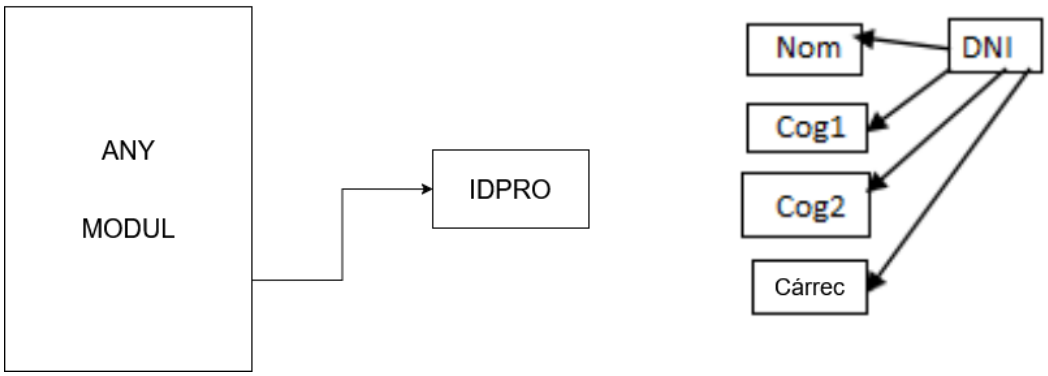
Las otras 2 tablas están en 2FN al solo tener un atributo como clave primaria sería imposible tener este problema.

3FN

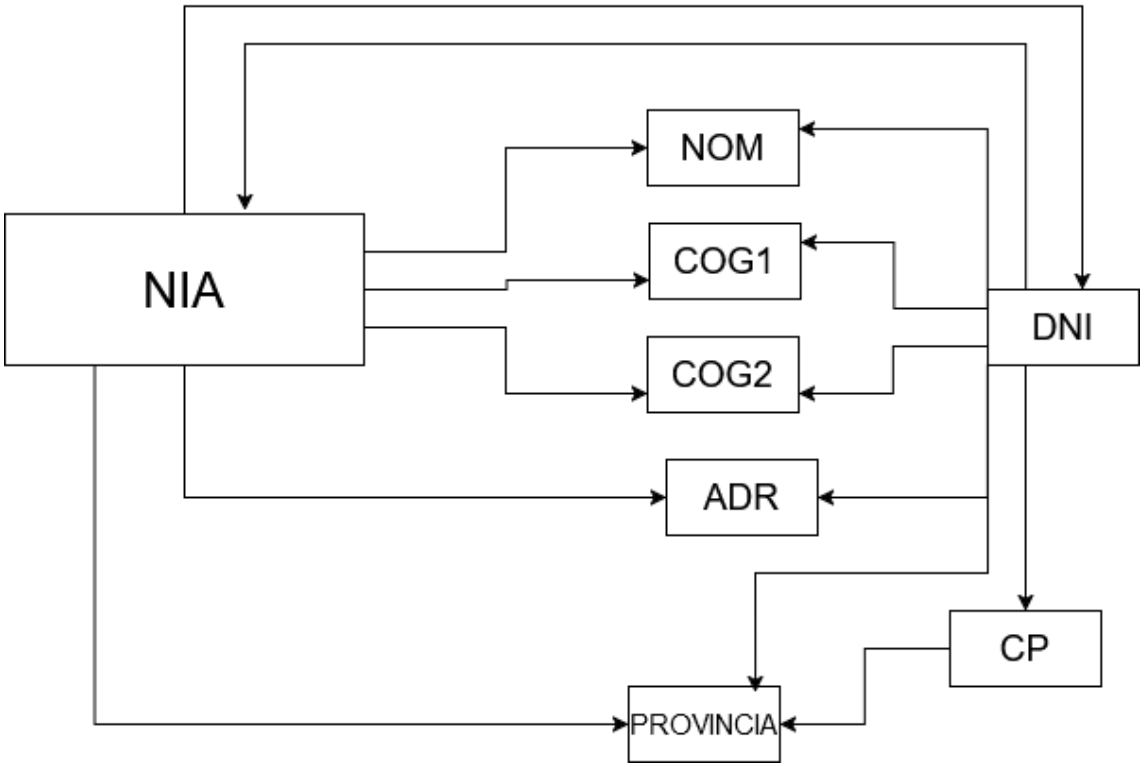
La tabla curso no esta en 3FN porque su atributo Cárrec depende transitivamente de la clave primaria.



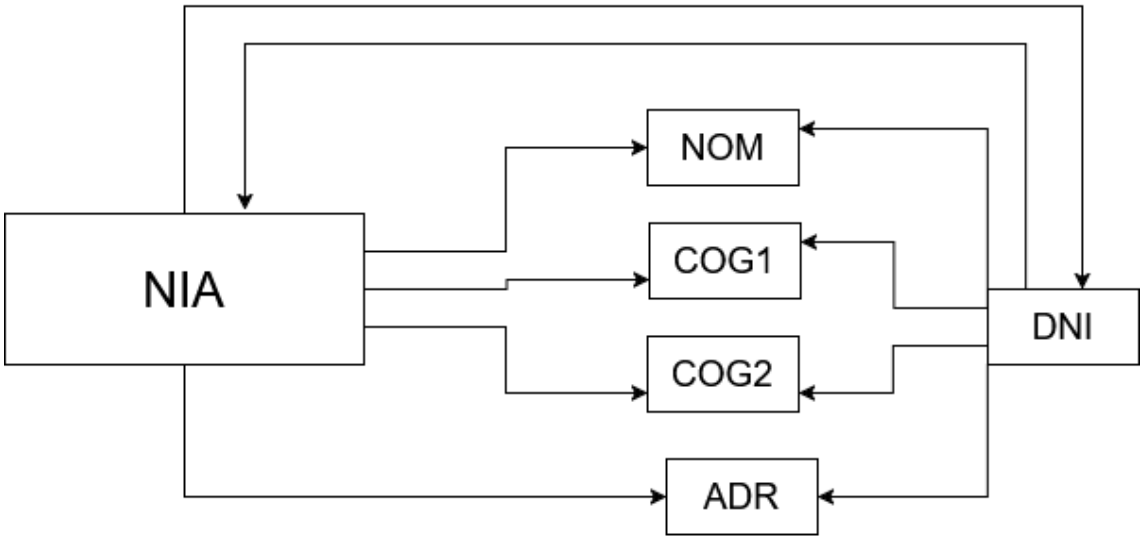
Para solucionar esto, hay que mandar al atributo a la tabla de la que depende. En este caso, depende de IdProfesor asique hay que mandarlo a la tabla profesor la cual ya existe de antes lo que nos dejaría las tablas curso y profesor así:

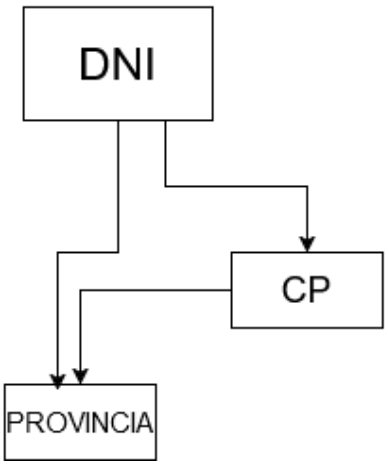


La tabla alumnos tampoco esta en 3FN ya que, aunque el resto de atributos tienen transitividades con 2 claves, CP y Provincia no.

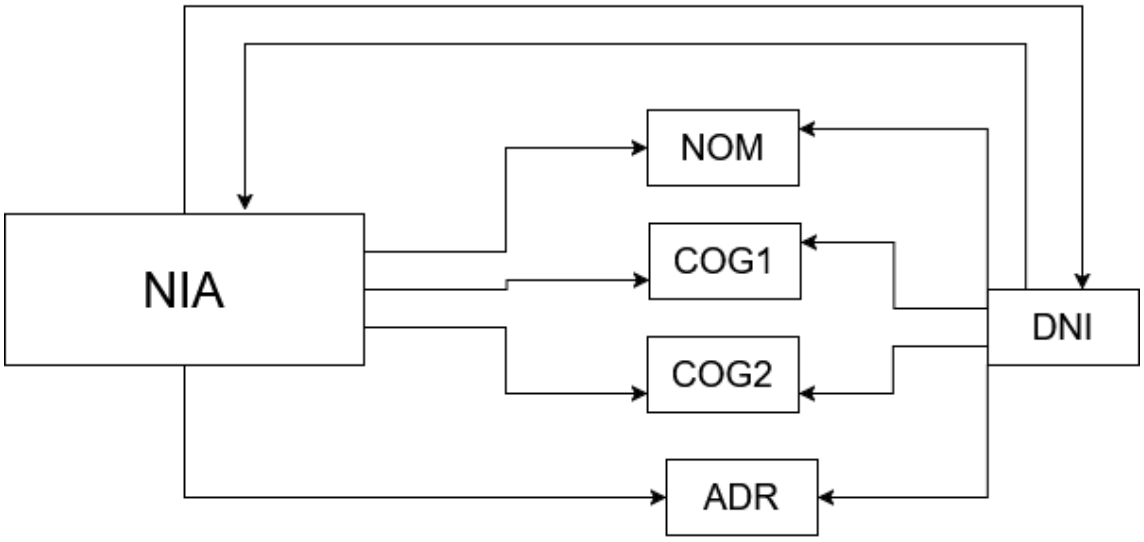


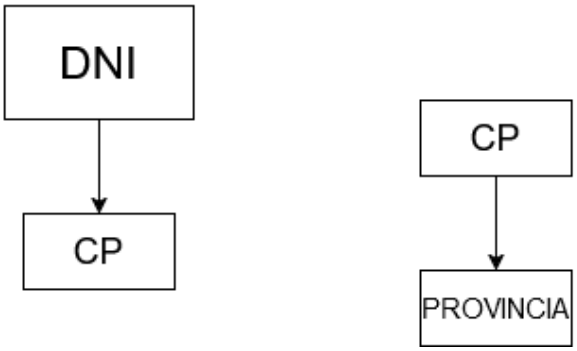
Por ende, hay que crear una nueva tabla con DNI como clave primaria para ahí poder guardar el código postal y la provincia. Nos quedaría la siguiente tabla a la que llamaremos persona:





Pero esta tabla tampoco esta en 3FN porque provincia es transitiva por código postal asique hay que volver a hacer otra tabla para códigos postales.





Al final el diseño lógico nos quedaría así:

ALUMNE(NIA, DNI, Nom, Cog1, Cog2, Adr)

CP:{ NIA }

UNI:{ DNI }

VNN:{ Nom, Cog1, Adr }

Caj:{DNI}->Persona

Persona(DNI, CP)

CP:{ DNI }

VNN:{CP}

Caj:{CP}->CodigoPostal

CodigoPostal(CP,Provincia)

CP:{CP}

VNN:{ Provincia }